1.     A digital data computing method comprising:

       executing a process that makes requests and that requires at least asynchronous responses to those requests to continue normal operation;

5

       generating those responses external to the process and supplying them to that process;

       the executing step including continuing normal operation of the process when at least asynchronous responses are received to the requests and otherwise discontinuing normal operation such that there is no real-time dependency of that process to those responses.

10

2.     The method of claim 1, comprising performing the executing step on a client and performing the generating step on a server.

3.     The method of claim 2, comprising performing the executing step on a server that comprises a secured coprocessor local to the client.

4.     The method of claim 2, comprising performing the executing step on a server that is remote with respect to the client.

5.     The method of claim 2, comprising performing the executing step utilizing a set of secured instructions and secured memory local to the client, where the instructions and memory are secured either by hardware or software.

25   6.     The method of claim 1, wherein it is computationally difficult to unauthorizedly simulate generation of the responses.

7.     The method of claim 6, wherein the executing step includes executing transformed code and wherein it is computationally difficult to determine proper responses to the requests without access to at least a portion of that code prior to a transformation that produces that transformed code.

8.      The method of claim 7, comprising performing the transformation automatically.

9.      The method of claim 7, comprising performing the transformation manually.

5

10.     The method of claim 1, wherein the generating step includes generating non-deterministic responses to the requests.

11.     The method of claim 10, wherein the executing step includes executing transformed code and wherein it is computationally difficult to generate the non-deterministic response without access to at least a portion of that code prior to a transformation that produces that transformed code.

12.     The method of claim 1, wherein the executing and generating steps are adapted to securing the generation of responses against any of unauthorized use, access, copying and functional analysis, and of controlling the execution of the process.

13.     A digital data computing method securing and controlling a set of instructions (hereafter, "code") against at least one of unauthorized use, access, copying and functional analysis comprising:

        including within the code requests to which the code requires at least asynchronous responses in order to continue normal operation;

        generating those responses external to the code and supplying them to that process;

        the executing step including continuing normal operation of the process when at least asynchronous responses are received to the requests and otherwise discontinuing normal operation such that there is no real-time dependency of that process to those responses.

14    The method of claim 13, wherein the code is comprised of high-level language or object code or any intermediary level set of computer instructions, or microcode.

15.    The method of claim 13, including the step of performing a transformation that includes generating any of code and data upon which the responses are based.

16.    The method of claim 15, comprising performing the transformation automatically.

17.    The method of claim 15, wherein performing the transformation manually.

18.    The method of claim 13, wherein it is computationally difficult to unauthorizedly simulate the generation of proper responses to the requests.

19.    The method of claim 18, wherein it is computationally difficult to generate the proper responses without access to at least a portion of code prior to the transformation.

20.    The method of claim 13, wherein the generating step includes a non-deterministic action.

21.    The method of claim 20, wherein it is computationally difficult to determine the effect of the non-deterministic action without access to at least a portion of the code prior to a transformation that produces that transformed code.

22.    The method of claim 15, comprising performing executing the code subsequent to transformation on a client and executing the generating step on a server.

23.    The method of claim 22, comprising performing the executing step on a server that comprises secured coprocessor local to the client.

24.    The method of claim 22, wherein the server is a remote processor.

25. The method of claim 22, wherein server is comprised of secured instructions utilizing secured memory local to the client, where the instructions and memory are secured either by hardware or software.

5   26. A digital data computing method, comprising:

executing a computer programming process, the executing step including performing any of allocation and de-allocation of data storage resources; and

10  providing data necessary for performing such allocation and de-allocation from a source external to the process, wherein that data includes at least one of a size and a location of an area to allocate or de-allocate.

27. The method of claim 26, wherein the executing step includes ceasing normal operation in the absence of such data from the external source within an expected delay interval.

15

28. The method of claim 26, wherein the executing step includes continuing normal operation in spite of at least expected delay of the data to the process.

20  29. The method of claim 26, comprising:

performing the executing step on a client;

generating the response data on a server; and

25

communicating the response data from the server to the client over a packet-switched network, local bus, local interface or other communications medium.

30. The method of claim 29, comprising performing the step of generating the response data on a sever that comprises any of a secured coprocessor and a hardware key.

30

31. The method of claim 30, wherein the server is a remote processor.

32. The method of claim 30, wherein server is comprised of secured instructions utilizing secured memory, where the instructions and memory are secured either by hardware or software.

33. A digital data computing method for securing and controlling the executing a set of instructions (hereafter, "code") against at least one of unauthorized use, access, copying and functional analysis comprising:

including, within the code, requests to which the code requires at least asynchronous responses in order to perform allocation and de-allocation of data storages resources;

generating those responses external to the process and supplying them to that process;

continuing normal operation of the code only if at least asynchronous responses are received to the requests and, otherwise, discontinuing normal operation, such that there is no real-time dependency of that code to those responses.

34. The method in claim 33, wherein the executing step includes ceasing normal operation in the absence of such data from the external source within an expected delay interval.

35. The method of claim 33, comprising:

performing the executing step on a client;

generating the response data on a server; and

communicating the response data from the server to the client over a packet-switched network, local bus, local interface or other communications medium.

36.     The method of claim 33, wherein the generating step includes a non-deterministic de-allocation action.

37.     A digital data computing method, comprising:

5

executing a computer programming process, the executing step including performing any of allocation and de-allocation of dynamic resources; and

providing data necessary for performing such allocation and de-allocation from a source

10      external to the process, wherein that data includes at least a description of the resource to be allocated or de-allocated.

38.     The method of claim 37, wherein the executing step includes ceasing normal operation in the absence of such data from the external source within an expected delay interval.

15

39.     The method of claim 37, wherein the executing step includes continuing normal operation in spite of at least expected delay of the data to the process.

40.     The method of claim 37, comprising:

20

performing the executing step on a client;

generating the response data on a server; and

25      communicating the response data from the server to the client over a packet-switched network, local bus, local interface or other communications medium.

41.     The method of claim 40, comprising performing the step of generating the response data on a server that is any of a coprocessor and a hardware key.

30

42.    The method of claim 40, comprising performing the step of generating the response data on a server that is a remote processor.

43.    The method of claim 40, wherein the step of generating the response data includes executing secured instructions utilizing secured memory, where the instructions and memory are secured either by hardware or software.